JOURNAL OF COMPUTER AND SYSTEM SCIENCES 28, 439–466 (1984)

A Complete Inference System for a Class of Regular Behaviours

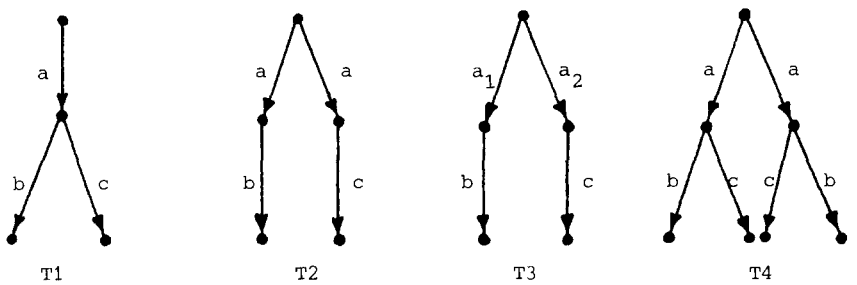
ROBIN MILNER

Department of Computer Science, University of Edinburgh, Edinburgh EH9 3JZ, Scotland

Received June 16, 1982; revised November 26, 1982

1. INTRODUCTION

The motivation of this paper is that a certain kind of tree, possibly infinite, is a useful model of computation. The trees we consider have arcs labelled by symbols taken from a set $ACT = \{a, b, ...\}$, which we call *actions*. Examples are:



Let us think of such a tree (in which we take no account of the order among the sons of a node) as modelling the behaviour of a machine accepting symbols from, or submitting to observations by, its environment. In tree T1 only the symbol a may be accepted at the start (or, only the observation a may be made), after which b or c may be accepted. The tree is determinate, since no node has two outgoing arcs with the same label. In T2 the same is true at the start, but the tree (or the machine modelled) is not determinate. Factors beyond the observer's control or knowledge will determine whether it is only b, or only c, which may be observed after the observation of a. Borrowing a phrase from C. A. Petri, we may think of information entering the observed system at different points in time—specifically, at each nondeterminate node of the tree. This information may be, for example, the influence of ambient temperature.

Alternatively, tree T2 may be a model suitable for an observer with imperfect vision, in that he cannot see a difference between actions a_1 and a_2, but takes them both to be a. For a better observer, the determinate tree T3 may be a better model. But even the first observer can discover that the machine modelled by T1 is different

0022-0000/84 $3.00

571/28/3-7

from that modelled by T2, since the former will always accept b after accepting a while the latter will sometimes refuse to do so.

This difference is represented in the model by a difference of trees; it is not represented when a machine is modelled by the set of (finite or infinite) sequences of symbols which it accepts, since these sets are identical for T1 and T2.

Sometimes, however, an observer may not be able to distinguish an indeterminate machine from a determinate one. The machines modelled by T4 and T1 may differ, but the information used by T4 to choose which branch to follow from its root is never used to effect a difference in future behaviour. Thus we argue that an observer will never distinguish these machines, and that they should be modelled by the same abstract object.

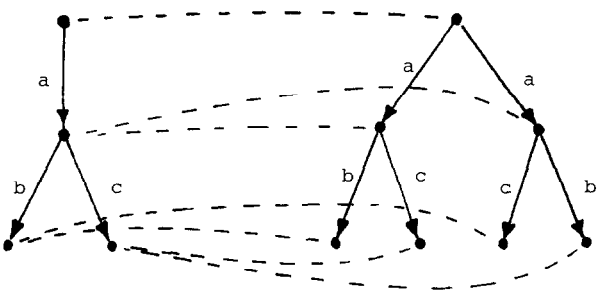
Since trees T4 and T1 differ (even when sons are not ordered), it follows that trees as they stand are not a suitable model. A possible refinement is to restrict attention to trees in which no node has two identical branches, where a branch is an action (arc label) paired with a son tree. This model would work perfectly for finite trees, but will not do in general; there are different trees of this kind which, from our view of behaviour, should not be distinguished (see Example 2 in Section 3 below).

The solution, however, is quite easy and mathematically tractable. We take as modelling objects not trees but certain equivalence classes of trees. (We do this implicitly already, when we say that order among sons is ignored.) The equivalence we use is that of *bisimulation*, a notion developed by Park [8] as a very fruitful improvement upon the observation equivalence studied by Hennessy and the author [4, 7]. For trees it is very straightforward. Let us write $n \xrightarrow{a} n'$ when node n of a tree leads to node n' by an arc labelled a. Then a bisimulation of trees T1 and T2 is a relation R between their node sets which contains the pair of roots, and such that if $\langle n_1, n_2 \rangle \in R$ then

(i) whenever $n_1 \xrightarrow{a} n_1'$ in T1, then for some n_2' in T2 $n_2 \xrightarrow{a} n_2'$ and $\langle n_1', n_2' \rangle \in R$;

(ii) whenever $n_2 \xrightarrow{a} n_2'$ in T2, then for some n_1' in T1 $n_1 \xrightarrow{a} n_1'$ and $\langle n_1', n_2' \rangle \in R$.

The illustrated trees T1 and T4 have a bisimulation, indicated by the broken lines in the diagram below:



We defer the formal details, though even now it should be rather obvious that we obtain an equivalence relation over trees in this way.

Now, recognising the need for taking the quotient of the set of trees by equivalence under bisimulation, we can widen the original set to include arbitrary transition graphs, since bisimulation can be just as well defined for them. Each resulting equivalence class is called the *behaviour* of its members. Part of the purpose of this paper is to place behaviours, so defined, on an equal footing with notions of behaviour (e.g., the languages of formal language theory) which are defined more directly by set-theoretic means.

A central result of the paper is a complete inference system for finitely presented behaviours, closely analogous to that of Salomaa [9] for regular sets of words. We begin in Section 2 with an algebra of transition graphs. Section 3 introduces the notion of bisimulation, and in Section 4 many properties of behaviours are derived with a view to establishing the soundness of our inference system. Section 5 sets up the inference system, and is mainly devoted to a proof of its completeness. In Section 6 we compare it closely with Salomaa's system; the analogy is so close that the significant differences emerge very clearly.

There are many relationships between this and other work, apart from those already mentioned. Many kinds of simulation exist (see Jensen [6] for example); frequently they are not symmetric, and bisimulation is a new notion as far as I am aware, though a special case of it is implicitly used in the standard procedure for reducing finite-state automata which can be found in any relevant textbook. Courcelle [2] has studied a class of infinite trees in depth; superficially at least the present definitions and approach are different and differently motivated, but more work is needed to establish the connection. The present material can probably, by a different treatment of variables, be fitted in to the framework of Elgot's [3] iterative algebraic theories. Finally, Two models for communicating processes provide an interesting contrast; the set-theoretic model of Hoare *et al.* [5] and that based upon metric spaces due to de Bakker and Zucker [1]. The present work arose from wishing to clarify my own approach [7] to the same topic by treating a special case, the observation of a single process, in depth.

2. Charts

We will introduce a kind of transition graph, called a *chart*, which is in one sense a generalisation of the familiar notion of a nondeterministic finite-state acceptor. In the latter, a node or state may be either accepting or nonaccepting. By contrast, we shall allow a node to be labelled by zero or more variables; a variable X indicates nodes at which the "behaviour" of the chart may be extended by substitution of another chart for X, in a way which will be made precise.

The significant departure from standard automata theory is in our treatment of the semantics of charts. We do not choose a language (set of words) as the meaning of a chart; instead, its meaning is taken to be a congruence class of charts under *bisimulation* (to be defined). Indeed, each such class will contain a tree-like chart—intuitively the unfolding of other charts in the class—and the traditional

language may be obtained as a certain subset of finite paths of this tree. But many charts equivalent in the standard sense are not congruent in our sense; the motivation for this more refined semantics has been discussed above.

In what follows, no sets are finite unless explicitly declared to be so.

We presuppose two fixed sets:

$$\text{Act} = \{a, a_1, ..., b, b_1, ...\}, \qquad \text{the } \textit{actions}$$
$$\text{Var} = \{X, X_1, ..., Y, Y_1, ...\}, \qquad \text{the } \textit{variables}.$$

DEFINITION. A *chart* C is a quadruple $\langle Q, s, D, E\rangle$ where:

Q is a nonempty set	(the nodes)
$s \in Q$	(the start node)
$D \subseteq Q \times \text{Act} \times Q$	(the derivations)
$E \subseteq Q \times \text{Var}$	(the extensions).

C is *finite* iff Q, D and E are finite.

We shall frequently write

$$D(q) = \{\langle a, q'\rangle \mid \langle q, a, q'\rangle \in D\}, \qquad \text{the derivations of } q$$
$$E(q) = \{X \mid \langle q, X\rangle \in D\}, \qquad \text{the extensions of } q.$$

Moreover, when C is understood we shall write

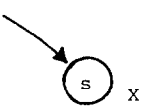
$$q \xrightarrow{a} q' \qquad \text{for} \quad \langle q, a, q'\rangle \in D$$
$$q \rhd X \qquad \text{for} \quad \langle q, X\rangle \in D.$$

In the former, we are treating D as the union of an indexed family $\{\xrightarrow{a} \mid a \in \text{Act}\}$ of binary relations over Q.

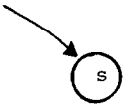
As is common practice, we shall not distinguish between charts $\langle Q_1, s_1, D_1, E_1\rangle$ and $\langle Q_2, s_2, D_2, E_2\rangle$ which are isomorphic, i.e., for which there is a bijection ϕ: $Q_1 \to Q_2$ such that $s_2 = \phi(S_1)$, $\langle q, a, q'\rangle \in D_1$ iff $\langle \phi(q), a, \phi(q')\rangle \in D_2$ and $\langle q, X\rangle \in E_1$ iff $\langle \phi(q), X\rangle \in E_2$.

We now turn to a simple algebra of charts; our algebraic operations will be seen later to respect the property of bisimulation, which justifies our use of the word congruence.

(1) *Variable.* For $X \in \text{Var}$, the chart $\langle \{s\}, s, \varnothing, \{\langle s, X\rangle\}\rangle$ is written X.



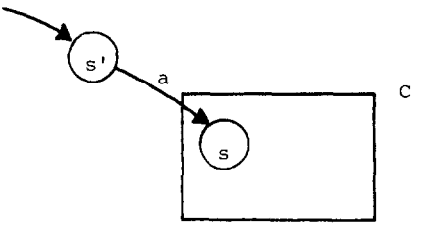
(2) *Null.* The null chart is $\mathbf{0} = \langle \{s\}, s, \varnothing, \varnothing \rangle$.



(3) *Prefix.* If $C = \langle Q, s, D, E \rangle$ and $a \in \text{Act}$, then

$$a(C) = \langle Q \cup \{s'\}, s', D \cup \{\langle s', a, s \rangle\}, E \rangle$$

where $s' \notin Q$.

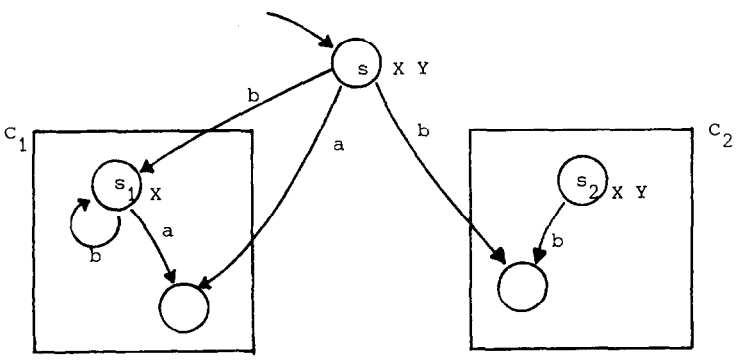


(4) *Sum.* Let $C_i = \langle Q_i, s_i, D_i, E_i \rangle$, $i \in \{1, 2\}$, where Q_1 and Q_2 are disjoint. The sum $C = C_1 + C_2$ is formed by adding a new node s which has the derivations and extensions of both s_1 and s_2. Formally, $C = \langle Q_1 \cup Q_2 \cup \{s\}, s, D_1 \cup D_2 \cup D', E_1 \cup E_2 \cup E' \rangle$, where

$$D' = \{s\} \times (D_1(s_1) \cup D_2(s_2))$$

$$E' = \{s\} \times (E_1(s_1) \cup E_2(s_2)).$$

EXAMPLE.



(5) *Substitution.* Let $\tilde{C} = (C_1, \ldots, C_n)$ and C be disjoint charts, and $\tilde{X} = (X_1, \ldots, X_n)$ distinct variables. The substitution $C' = C[\tilde{C}/\tilde{X}]$ is formed by replacing each extension X_i in C by the derivations and extensions of the start node s_i in C_i.

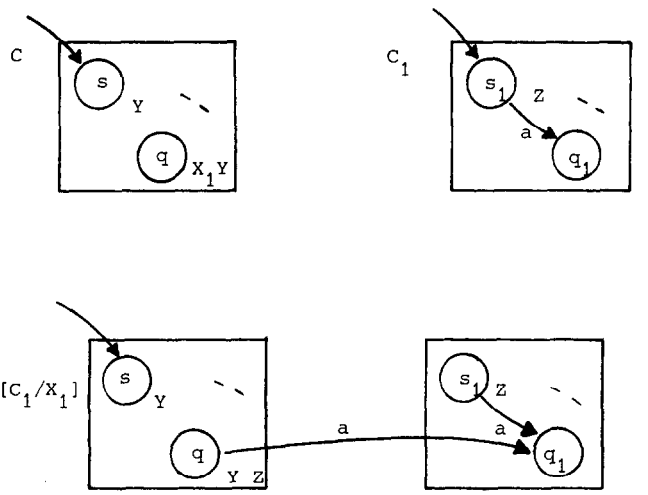
Formally,

$$C' = \langle Q \cup UQ_i, s, D' \cup UD_i, E' \cup UE_i \rangle$$

where for $q \in Q_i$, $D'(q) = E'(q) = \varnothing$, while for $q \in Q$

$$D'(q) = D(q) \cup U\{D_i(s_i) \mid X_i \in E(q)\}$$
$$E'(q) = E(q) - \tilde{X} \cup U\{E_i(s_i) \mid X_i \in E(q)\}.$$

EXAMPLE.



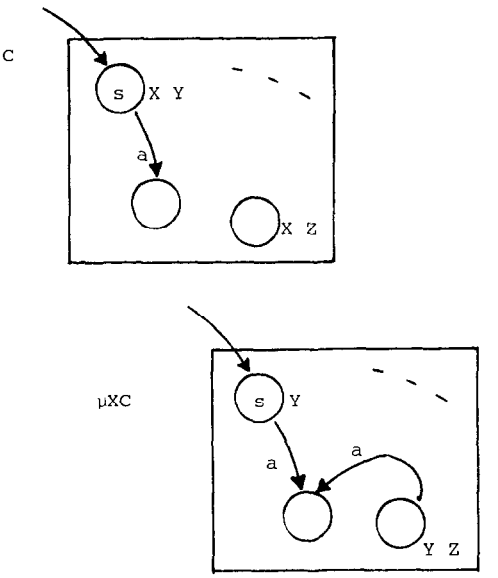
It may be helpful to remark here that the start-node of C_1, rendered inaccessible in this construction, causes no problem; the effect of bisimulation will be to ignore its presence.

(6) *Recursion.* If $C = \langle Q, s, D, E \rangle$, then the recursion μXC is formed by replacing each extension X in C by the derivations and extensions (except X) of s in C. Formally $\mu XC = \langle Q, s, D^+, E^+ \rangle$, where

$$D^+(q) = \begin{cases} D(q) \cup D(s) & \text{if} \quad X \in E(q) \\ D(q) & \text{otherwise} \end{cases}$$

$$E^+(q) = \begin{cases} E(q) \cup E(s) - \{X\} & \text{if} \quad X \in E(q) \\ E(q) & \text{otherwise.} \end{cases}$$

EXAMPLE.



Note that the presence or absence of X as an extension of s makes no difference, and that X is not an extension at all in μXC.

To summarise:

Null $\mathbf{0}$ and Variable X are nullary chart operations;

Prefix $a(-)$ and $\mu X(-)$ are unary operations;

Sum $+$ is a binary operation;

Substitution $(-)[-/\tilde{X}]$ is an $(n+1)$-ary operation for each n-vector $\tilde{X}$ of distinct variables.

An n-ary operation for mutual recursion, $\mu_i \tilde{X}\tilde{C}$, can also be defined, where $\tilde{X}$ is an n-vector of distinct variables and $\tilde{C}$ an n-vector of charts; intuitively, it stands for the ith component of a solution of the equations $\tilde{X} = \tilde{C}$. Indeed, infinitary versions of recursion, substitution and sum are easily defined, and have some significance for broader work. Here we shall be mainly concerned with finitely presentable charts, for which indeed unary recursion has sufficient expressive power.

We should notice that a finite chart in which at most one variable occurs is not quite the standard nondeterministic finite-state acceptor; we have not included empty transitions (which may occur unobserved, i.e., which accept no input symbol). More is said on this point at the end of Section 5.

3. BISIMULATION AND CONGRUENCE

Following Park [8], and by analogy with the strong congruence of Milner [7], we approach the notion of behaviour of a chart as follows.

DEFINITION. Let $C_i = \langle Q_i, s_i, D_i, E_i \rangle$ be charts, $i = 1, 2$. Then $R \subseteq Q_1 \times Q_2$ is a *bisimulation* of C_1 *and* C_2 if

(1) $\langle s_1, s_2 \rangle \in R$,

(2) $\langle q_1, q_2 \rangle \in R$ implies

(i) For every $q_1 \xrightarrow{a} q_1'$ in D_1, there exists $q_2 \xrightarrow{a} q_2'$ in D_2 for which $\langle q_1', q_2' \rangle \in R$;

(ii) For every $q_2 \xrightarrow{a} q_2'$ in D_2, there exists $q_1 \xrightarrow{a} q_1'$ in D_1 for which $\langle q_1', q_2' \rangle \in R$;

(iii) $E_1(q_1) = E_2(q_2)$.

Loosely speaking, condition (2) says that q_1 and q_2 must have equal derivations (up to R) and equal extensions. Also, if we denote by $\mathscr{F}(R)$ the set of pairs $\langle q_1, q_2 \rangle$ satisfying clauses (i)–(iii), then condition (2) may be rewritten simply

$$R \subseteq \mathscr{F}(R).$$

This formulation allows a clean mathematical treatment, arising from the fact that the function $\mathscr{F}$ over relations is monotone (for inclusion of relations) and preserves many simple properties of relations, such as reflexivity, symmetry and transitivity. But for our present purposes we can work directly from our definition as it stands.

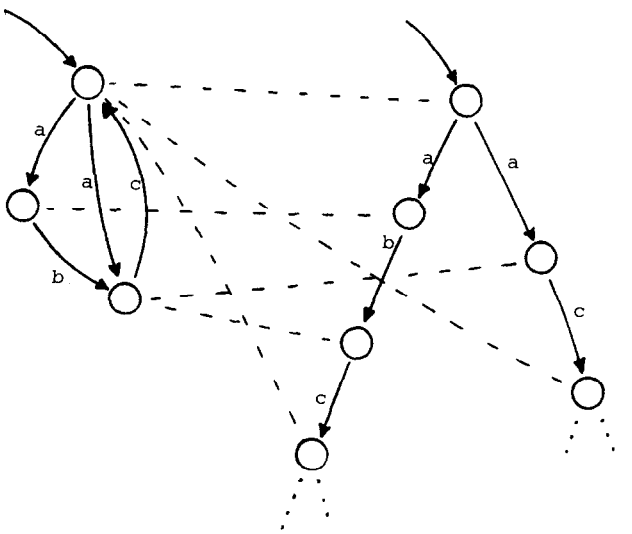
DEFINITION. Charts C_1 and C_2 are *congruent* if they possess a bisimulation; we write $C_1 \sim C_2$.

PROPOSITION 3.1. *Congruence of charts is an equivalence relation.*

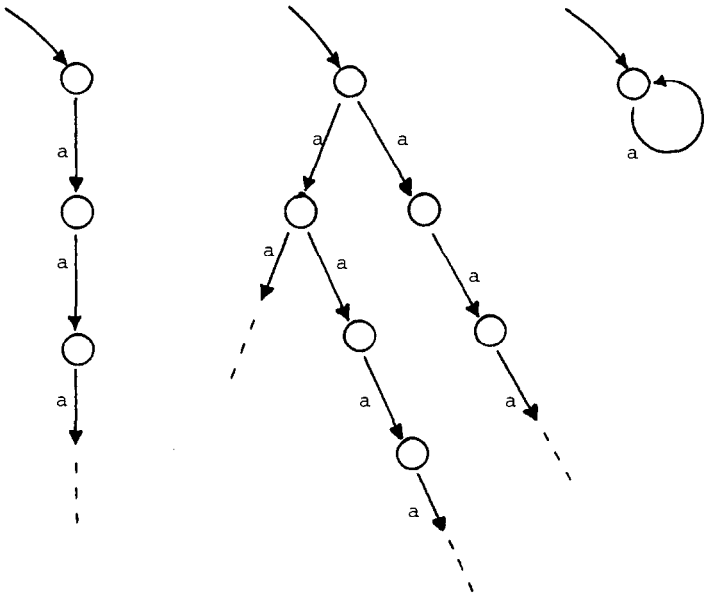
Proof. Immediate from the definition. ∎

Some simple examples illustrate the notion.

(1) Every chart may be unfolded, from the start-node onwards, into a congruent tree-like chart. In the diagram, the bisimulation is indicated by broken lines:

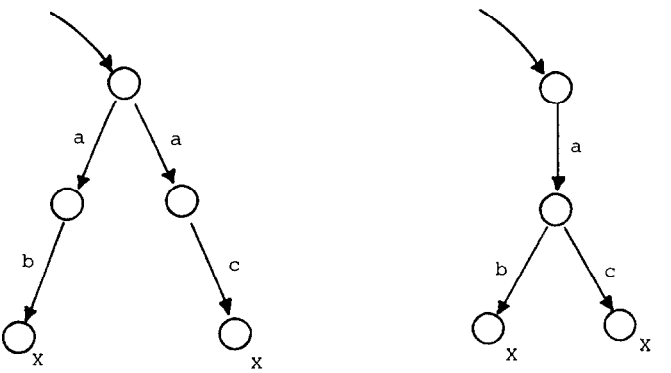


(2) Even distinct tree-like charts may be congruent; the two below are trivially congruent to a single-node chart.



Our excuse for confronting the reader with this "triviality" is that it shows why it would not be enough to take tree-charts, under any reasonably simple definition, as our semantic objects; we should still require congruence classes of them.

(3) Two finite charts, equivalent as acceptors in the standard sense of automata theory but not congruent, are shown below (think of nodes labelled X as accept states):



We shall often write $\tilde{C} \sim \tilde{C}'$ when $\tilde{C} = C_1,..., C_n$ and $\tilde{C}' = C_1',..., C_n'$ and $C_i \sim C_i'$ $(i \leqslant n)$. We now show that our chart operations respect congruence (justifying the term).

PROPOSITION 3.2. *Let* $C \sim C'$, $C_1 \sim C_1', \ldots, C_n \sim C_n'$. *Then*

(1) $aC \sim aC'$,

(2) $C_1 + C_2 \sim C_1' + C_2'$,

(3) $C[\tilde{C}/\tilde{X}] \sim C'[\tilde{C}'/\tilde{X}]$,

(4) $\mu XC \sim \mu XC'$.

Proof. In each case a bisimulation can be constructed from bisimulations R for C and C', and R_i for C_i and C_i'. The details of proof are completely routine from the definitions of the operations. In (2), for example, if s and s' are the new nodes added in the respective sums, then $R_1 \cup R_2 \cup \{\langle s, s' \rangle\}$ is the required bisimulation; in (3) it is simply $R \cup R_1 \cup \cdots \cup R_n$.

∎

Before turning to the algebra of congruence classes, we need to establish the important property that recursion is (up to congruence) a fixed-point for substitution.

PROPOSITION 3.3 (FIXED-POINT).

$$\mu XC \sim C[\mu XC/X].$$

Proof. Let $C = \langle Q, s, D, E \rangle$, so that $\mu XC = \langle Q, s, D^+, E^+ \rangle$, where D^+, E^+ are as given in Section 2. Let $C' = \langle Q', s', D', E' \rangle$ be isomorphic to C, under the bijection $\phi: q \mapsto q'$, with Q and Q' disjoint, and let $C'' = C'[\mu XC/X]$. We have to show

$$\mu XC \sim C''.$$

Now $C'' = \langle Q' \cup Q, s', D'', E'' \rangle$, where from definitions we can deduce

(i) If $q' \in Q'$ then

$$D''(q') = \begin{cases} D'(q') \cup D(s) & \text{if} \quad X \in E'(q') \\ D'(q') & \text{otherwise} \end{cases}$$

$$E''(q') = \begin{cases} E'(q') \cup E(s) - \{X\} & \text{if} \quad X \in E'(q') \\ E'(q') & \text{otherwise;} \end{cases}$$

(ii) If $q \in Q$ then

$$D''(q) = D^+(q) = \begin{cases} D(q) \cup D(s) & \text{if} \quad X \in E(q) \\ D(q) & \text{otherwise} \end{cases}$$

$$E''(q) = E^+(q) = \begin{cases} E(q) \cup E(s) - \{X\} & \text{if} \quad X \in E(q) \\ E(q) & \text{otherwise.} \end{cases}$$

Now it is easy to check that $\mathscr{R} = \mathrm{Id}_Q \cup \phi$ is a bisimulation of μXC and C''. Certainly $\langle s, s' \rangle \in R$; the remaining conditions are easily established from (i) and (ii) bearing

in mind that, since C and C' are isomorphic, $E'(q') = E(q)$ and $D'(q') = \{\langle a, p'\rangle \mid \langle a, p\rangle \in D(q)\}$. ∎

PROPOSITION 3.4.

$$\mu X(C + X) \sim \mu X C.$$

Proof. If $C = \langle Q, s, D, E\rangle$, then let s' be the start node of $C + X$, hence also of $\mu X(C + X)$. It is readily seen that $\mathrm{Id}_Q \cup \{\langle s', s\rangle\}$ is a bisimilation of $\mu X(C + X)$ and $\mu X C$. ∎

4. BEHAVIOURS AND THEIR PROPERTIES

DEFINITION. A *behaviour* is a congruence class of charts.

We will denote the set of behaviours by $\mathscr{B}$. Having already established that our chart operations are well defined on $\mathscr{B}$, we also define derivations and extensions of behaviours.

DEFINITION. The relations $\mathscr{D} \subseteq \mathscr{B} \times \mathrm{Act} \times \mathscr{B}$ and $\mathscr{E} \subseteq \mathscr{B} \times \mathrm{Var}$ are given as follows:

(1) $\langle B, a, B'\rangle \in \mathscr{D}$ (written $B \xrightarrow{a} B'$) iff, for some $C = \langle Q, s, D, E\rangle$ in B, there exists s' in Q such that $\langle s, a, s'\rangle \in D$ and $\langle Q, s', D, E\rangle \in B'$;

(2) $\langle B, X\rangle \in \mathscr{E}$ (written $B \rhd X$) iff for some $C = \langle Q, s, D, E\rangle$ in B, $\langle s, X\rangle \in E$.

It is readily established that, in each case, if *some* C in B has the stated property, then *every* C in B has it.

Thus each behaviour B may be regarded as a chart $\langle \mathscr{B}, B, \mathscr{D}, \mathscr{E}\rangle$; in this "universal" chart a behaviour is identified with a choice of start-node, and we may therefore define congruence of behaviours:

$$B_1 \sim B_2 \quad \text{iff} \quad \langle \mathscr{B}, B_1, \mathscr{D}, \mathscr{E}\rangle \sim \langle \mathscr{B}, B_2, \mathscr{D}, \mathscr{E}\rangle.$$

But our universal chart is suitably abstract; in fact, congruent behaviours are equal!

PROPOSITION 4.1. *If* $B_1 \sim B_2$ *then* $B_1 = B_2$.

Proof. Let $\mathscr{R} \subseteq \mathscr{B} \times \mathscr{B}$ be a bisimulation containing $\langle B_1, B_2\rangle$. Now choose $C_1 = \langle Q_1, s_1, D_1, E_1\rangle \in B_1$ and $C_2 = \langle Q_2, s_2, D_2, E_2\rangle \in B_2$. We require $C_1 \sim C_2$; for this purpose we show that $R \subseteq Q_1 \times Q_2$, defined by

$$R = \{\langle q_1, q_2\rangle \mid \langle Q_1, q_1, D_1, E_1\rangle \in A_1, \langle Q_2, q_2, D_2, E_2\rangle \in A_2, \langle A_1, A_2\rangle \in \mathscr{R}\}$$

is a bisimulation of C_1 and C_2. Certainly $\langle s_1, s_2\rangle \in R$, by taking $A_1, A_2 = B_1, B_2$. Now suppose $\langle q_1, q_2\rangle \in R$.

(i) Let $q_1 \xrightarrow{a} q_1'$ in D_1. We require q_2' such that $q_2 \xrightarrow{a} q_2'$ in D_2 and $\langle q_1', q_2' \rangle \in R$. But $A_1 \xrightarrow{a} A_1'$, where $\langle Q_1, q_1', D_1, E_1 \rangle \in A_1'$; hence $A_2 \xrightarrow{a} A_2'$, with $\langle A_1', A_2' \rangle \in R$; then also $q_2 \xrightarrow{a} q_2'$ in D_2, with $\langle Q_2, q_2', D_2, E_2 \rangle \in A_2'$, and finally $\langle q_1', q_2' \rangle \in R$.

(ii) Let $q_1 \rhd X$ in E_1. Then $A_1 \rhd X$, so $A_2 \rhd X$, so $q_2 \rhd X$ in E_2.

It follows, by symmetric argument, that R is a bisimulation, and from $C_1 \sim C_2$ it follows that $B_1 = B_2$. ∎

Remark. We shall use this proposition to establish equational properties of behaviours.

To reason directly about behaviours, we need to relate the derivatives and extensions of the result of an operation to those of the operands. This is expressed in the following proposition, which we state without proof.

PROPOSITION 4.2 (BEHAVIOUR DERIVATIONS AND EXTENSIONS).

(1) $\mathscr{D}(\mathbf{0}) = \varnothing$, $\mathscr{E}(\mathbf{0}) = \varnothing$.

(2) $\mathscr{D}(X) = \varnothing$, $\mathscr{E}(X) = \{X\}$.

(3) $\mathscr{D}(aB) = \{\langle a, B \rangle\}$, $\mathscr{E}(aB) = \varnothing$.

(4) $\mathscr{D}(B_1 + B_2) = \mathscr{D}(B_1) \cup \mathscr{D}(B_2)$, $\mathscr{E}(B_1 + B_2) = \mathscr{E}(B_1) \cup \mathscr{E}(B_2)$.

(5) $\mathscr{D}(B[\tilde{A}/\tilde{X}]) = \{\langle a, B'[\tilde{A}/\tilde{X}] \rangle \mid \langle a, B' \rangle \in \mathscr{D}(B)\} \cup U\{\mathscr{D}(A_i) \mid X_i \in \mathscr{E}(B)\}$, $\mathscr{E}(B[\tilde{A}/\tilde{X}]) = \mathscr{E}(B) - \tilde{X} \cup U\{\mathscr{E}(A_i) \mid X_i \in \mathscr{E}(B)\}$.

(6) *If* $\mathscr{D}(B_1) = \mathscr{D}(B_2)$ *and* $\mathscr{E}(B_1) = \mathscr{E}(B_2)$ *then* $B_1 = B_2$.

We may now begin to establish equational properties from which will follow the soundness of the inference system to be presented in the following section.

PROPOSITION 4.3 (SUM).

(1) $A + B = B + A$.

(2) $A + (B + C) = (A + B) + C$.

(3) $A + A = A$.

(4) $A + \mathbf{0} = A$.

Proof. Directly from Proposition 4.2(1), (4), (6). ∎

The following properties are crucial for inference.

PROPOSITION 4.4 (RECURSION).

(1) $\mu XB = B[\mu XB/X]$.

(2) $\mu X(B + X) = \mu XB$.

(3) *If* $A = B[A/X]$ *and* $B \not\rhd X$ *then* $A = \mu XB$.

Proof. (1) From Proposition 3.3.

(2) From Proposition 3.4.

(3) By (1), it will be enough to show that if $A_1 = B[A_1/X]$ and $A_2 = B[A_2/X]$ then $A_1 = A_2$.

Let $\mathscr{R} = \{\langle C[A_1/X], C[A_2/X]\rangle \mid C \in \mathscr{B}\}$. We wish to show that $\mathscr{R}$ is a bisimilation of the behaviours A_1 and A_2. Clearly $\langle A_1, A_2\rangle \in \mathscr{R}$ (take $C = B$), and it remains to prove that $\mathscr{R} \subseteq \mathscr{F}(\mathscr{R})$.

For arbitrary C, let $C[A_1/X] \xrightarrow{a} D_1$. Then from Proposition 4.2(5) *either*

(i) $C \xrightarrow{a} C'$ and $D_1 = C'[A_1/X]$; but then also $C[A_2/X] \xrightarrow{a} D_2 = C'[A_2/X]$, and $\langle D_1, D_2\rangle \in \mathscr{R}$, *or*

(ii) $C \rhd X$ and $A_1 \xrightarrow{a} D_1$; then $B[A_1/X] \xrightarrow{a} D_1$, whence (since not $B \rhd X$) $B \xrightarrow{a} B'$ and $D_1 = B'[A_1/X]$; also $A_2 = B[A_2/X] \xrightarrow{a} D_2 = B'[A_2/X]$ and $\langle D_1, D_2\rangle \in \mathscr{R}$; further, $C[A_2/X] \xrightarrow{a} D_2$ also since $C \rhd X$.

Now let $C[A_1/X] \rhd Y$. It follows by a similar (easier) argument that $C[A_2/X] \rhd Y$.

Hence, by symmetric argument, $\mathscr{R}$ is a bisimulation, and the result follows, by Proposition 4.1. ∎

It is convenient to define a few simple notions before the last proposition of this section, which presents properties of substitution.

DEFINITION. A *computation* of a behaviour B_0 is either of form

$$B_0 \xrightarrow{a_1} B_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} B_n = \mathbf{0}, \qquad n \geqslant 0 \quad (\text{a } \textit{finite} \text{ computation})$$

or of form

$$B_0 \xrightarrow{a_1} B_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} B_n \xrightarrow{a_{n+1}} \cdots \qquad (\text{an } \textit{infinite} \text{ computation})$$

Then each B_i $(i \geqslant 0)$ is a *derivative*, and each B_i $(i \geqslant 1)$ a *proper derivative*, of B_0.

DEFINITION. X is *live in* B if $B' \rhd X$ for some derivative B' of B; otherwise X is *dead in* B.

The following, whose simple proof we omit, is sometimes useful.

PROPOSITION 4.5. *X is live in B iff every chart in B has a node with extension X.*

Finally, we give the properties of substitution.

PROPOSITION 4.6.

(1) $\mathbf{0}[\tilde{A}/\tilde{X}] = \mathbf{0}$.

(2) $Y[\tilde{A}/\tilde{X}] = \begin{cases} A_i & \text{if } Y \equiv X_i \\ Y & \text{if } Y \notin \tilde{X}. \end{cases}$

(3) $(aB)[\tilde{A}/\tilde{X}] = a(B[\tilde{A}/\tilde{X}])$.

(4) $(B_1 + B_2)[\tilde{A}/\tilde{X}] = B_1[\tilde{A}/\tilde{X}] + B_2[\tilde{A}/\tilde{X}]$.

(5) *If* Z *is dead in* μYB *then* $\mu YB = \mu Z(B[Z/Y])$.

(6) *If* Y *is dead in* $\tilde{A}$ *and not in* $\tilde{X}$ *then* $(\mu YB)[\tilde{A}/\tilde{X}] = \mu Y(B[\tilde{A}/\tilde{X}])$.

Proof. In each case, the corresponding congruences for charts may be shown by simple bisimulations of charts. Alternatively, Proposition 4.2 may be used, at least for (1)–(4). For (5), we may also use bisimulation of behaviours, as follows. Let A_1 and A_2 be the two sides of the equation. Then $A_1 = B[A_1/Y]$ and $A_2 = B[Z/Y][A_2/Z]$. The relation $\{\langle C[A_1/Y], C[Z/Y][A_2/Z]\rangle \mid C \in \mathscr{B}\}$ may then be shown, using Proposition 4.2(5), to be a bisimulation of A_1 and A_2. ∎

Having now listed many properties of behaviours, we may ask which are the important ones. It turns out that this question may be answered firmly if we are only interested in finitely expressible behaviours, i.e., those for which we can find an expression in terms of our operations. In fact, the next section provides a complete inference system for such behaviours.

5. Expressions and Inference

The expressions E of our inference system are those which may be built with the symbols we have been using for behaviour operations, excluding substitution, and allowing parentheses wherever necessary or convenient. Thus, using conventional notation for syntax definition,

$$E ::= \mathbf{0} \mid X \mid aE \mid E + E \mid \mu XE \qquad (X \in \text{Var}, a \in \text{Act}).$$

Though we now need to discuss syntax as well as what it stands for, we shall not find it necessary to make explicit the "semantic function" which takes an expression to its meaning (a behaviour); the context will make it clear whether an expression or its meaning is intended, together with the convention that "$\equiv$" is used for identity of expressions and "$=$" for equality of meaning. Also, we shall use E, F, G for expressions, and A, B, C for behaviours (which may not always be expressible).

Definition. B is *regular* if it has a finite chart.

Note that this definition agrees with that normally given for trees (a tree is regular if it has finitely many distinct subtrees), as it is easily seen that B is regular iff it has finitely many distinct derivatives and employs only a finite part of Act and Var.

Definition. B is *expressible* if it is the meaning of some expression.

Proposition 5.1. *If* B *is expressible then it is regular.*

Proof. By induction on expressions. ∎

The converse will follow as a corollary of Theorem 5.7.

DEFINITION. X is *free* in expression E if E contains an occurrence of X not contained in any subexpression μXF. X is *guarded* in expression E if every free occurrence of X in E is contained in a subexpression aF.

EXAMPLES. X is guarded in aX, μXX and $Y + \mathbf{0}$, but not in X, $aX + X$ or $\mu Y(aY + X)$.

Our next task is to relate these syntactic concepts to semantic ones. Fortunately, it is very simple.

PROPOSITION 5.2. *X is guarded in E iff $E \not\rhd X$.*

Note. In this sentence, the first E stands for an expression, the second E for its meaning.

Proof. By induction on E. We only need consider briefly the case μYE. If X is guarded in μYE, then *either* $X \equiv Y$, and certainly $\mu XE \not\rhd X$ (by definition of μX on charts), *or* $X \not\equiv Y$ and X is guarded in E, whence by induction $E \not\rhd X$, and again it follows that $\mu XE \not\rhd X$. If X is unguarded in μYE then $X \not\equiv Y$ and X is unguarded in E, so $E \rhd X$, whence $\mu YE = E[\mu YE/Y] \rhd X$ by Proposition 4.2(5). ∎

PROPOSITION 5.3. *X is free in E iff X is live in E.*

Proof. By induction on E. Consider only the case μYE. For clarity, let E denote behaviour B. If X is free in μYE then $X \not\equiv Y$ and X is free in E, so X is live in B, and $B' \rhd X$ for some derivative B' of B. But then $\mu YB = B[\mu YB/Y]$ has a derivative $B'[\mu YB/Y]$ (by iterative use of Proposition 4.2(5)) $\rhd X$; hence X is live in μYE.

If X is not free in μYE, then *either* $X \equiv Y$, and then μYB has a chart with no extension X, so X is dead in μYE by Proposition 4.5, *or* $X \not\equiv Y$ and X is not free in E, so not live in B, and again (with the help of Proposition 4.5) μYB has a chart with no extension X, so X is dead in μYE. ∎

To formulate our axiom system we need—as in the lambda calculus—a definition of syntactic substitution. $F\{\tilde{E}/\tilde{X}\}$ stands for the simultaneous substitution of expressions $\tilde{E}$ for variables $\tilde{X}$ in expression F. $-\{-/\tilde{X}\}$ is thus an operation on expressions, while $-[-/\tilde{X}]$ operates on behaviours.

DEFINITION. $F\{\tilde{E}/\tilde{X}\}$ is defined by induction on F as follows:

$$\mathbf{0}\{\tilde{E}/\tilde{X}\} \equiv \mathbf{0}$$

$$Y\{\tilde{E}/\tilde{X}\} \equiv Y \quad \text{if} \cdot Y \notin \tilde{X}$$

$$\equiv E_i \quad \text{if} \quad Y \equiv X_i$$

$$aF\{\tilde{E}/\tilde{X}\} \equiv a(F\{\tilde{E}/\tilde{X}\})$$

$$(F+G)\{\tilde{E}/\tilde{X}\} \equiv F\{\tilde{E}/\tilde{X}\} + G\{\tilde{E}/\tilde{X}\}$$

$$(\mu YF)\{\tilde{E}/\tilde{X}\} \equiv \mu Y(F\{\tilde{E}/\tilde{X}\}) \quad \text{if} \quad Y \text{ is not in } \tilde{X} \text{ nor free in } \tilde{E};$$

$$\equiv \mu Z(F\{Z/Y\}\{\tilde{E}/\tilde{X}\}) \quad \text{otherwise, for some } Z \text{ not in } \tilde{X} \text{ nor free in } \mu YF \text{ or } \tilde{E}.$$

The following essential proposition, relating syntactic and semantic substitution, can now be proved.

PROPOSITION 5.4. $F\{\tilde{E}/\tilde{X}\} = F[\tilde{E}/\tilde{X}]$.

Proof. By induction on the number of operation symbols in F. We consider only the case μYF, in the case where Y is in $\tilde{X}$ or free in $\tilde{E}$.

On the one hand, $F\{\tilde{E}/\tilde{X}\} \equiv \mu Z(F\{Z/Y\}\{\tilde{E}/\tilde{X}\})$, with Z not in $\tilde{X}$ and not free in μYF or $\tilde{E}$. But, since $F\{Z/Y\}$ has exactly the same number of operation symbols as F, by induction $F\{Z/Y\}\{\tilde{E}/\tilde{X}\} = F\{Z/Y\}[\tilde{E}/\tilde{X}]$ (semantic equality) and also $F\{Z/Y\} = F[Z/Y]$; hence (by congruence) $F\{\tilde{E}/\tilde{X}\} = \mu Z(F[Z/Y][\tilde{E}/\tilde{X}])$.

On the other hand, using Proposition 5.3 (free = live) and Proposition 4.6(5), $\mu YF = \mu Z(F[Z/Y])$, and by Proposition 4.6(6) $\mu Z(F[Z/Y]) = \mu Z(F[Z/Y][\tilde{E}/\tilde{X}])$. ∎

We are now ready to present our inference system, whose sentences are equations $E = F$ (we occasionally abbreviate a finite sequence of equations by $\tilde{E} = \tilde{F}$).

Rules of Inference

Equivalence:	E1.	$E = E$
	E2.	From $E = F$ infer $F = E$
	E3.	From $E = F$ and $F = G$ infer $E = G$
Congruence:	C1.	From $\tilde{E} = \tilde{E}'$ and $F = F'$ infer $F\{\tilde{E}/\tilde{X}\} = F'\{\tilde{E}'/\tilde{X}\}$
	C2.	From $E = E'$ infer $\mu XE = \mu XE'$
Summation:	S1.	$E + F = F + E$
	S2.	$E + (F + G) = (E + F) + G$
	S3.	$E + E = E$
	S4.	$E + \mathbf{0} = E$
Recursion:	R1.	$\mu XE = \mu Y(E\{Y/X\})$, Y not free in μXE
	R2.	$\mu XE = E\{\mu XE/X\}$
	R3.	$\mu X(E + X) = \mu XE$
	R4.	From $E = F\{E/X\}$, X guarded in F, infer $E = \mu XF$.

Notes. (i) We have not been concerned with reducing the equivalence and congruence rules to a minimum; some smaller set of rules may be equipotent.

(ii) It is significant that we need no explicit rule for prefixes aE. In particular, the distributive law $a(E + F) = aE + aF$ is not valid.

If an equation $E = F$ may be deduced by the rules, we write $\vdash E = F$.

THEOREM 5.5 (SOUNDNESS). *If* $\vdash E = F$ *then* $E = F$.

Proof. As usual, we must show that each rule of inference is valid. For (E1)–(E3), (C1)–(C2) and (S1)–(S4) appeal to Propositions 3.1, 3.2 and 4.3, respectively. For (R1)–(R4) appeal to Propositions 4.6(5) and 4.4, using Propositions 5.2–5.4 to relate syntactic and semantic notions. ∎

The converse, completeness, will be proved in Theorem 5.10. In it and the theorems leading to it, we need to appeal to provable properties of syntactic substitution. The congruence rules are often used without explicit reference, but we also need the following technical lemma.

LEMMA 5.6. (1) *If no* X_i *is free in* E *then* $\vdash E\{\tilde{F}/\tilde{X}\} = E$.

(2) *If* $\tilde{X}$ *and* $\tilde{Y}$ *are disjoint then*

$$\vdash E\{\tilde{F}/\tilde{X}\}\{\tilde{G}/\tilde{Y}\} = E\{\tilde{F}\{\tilde{G}/\tilde{Y}\}/\tilde{X}, \tilde{G}/\tilde{Y}\}.$$

Proof. By induction on the size of E. The only point of interest arises in the case μZE; in (2), if Z is in $\tilde{X}$ or $\tilde{Y}$ or free in $\tilde{F}$ or $\tilde{G}$, one must first use $\vdash \mu ZE = \mu V(E\{V/Z\})$ (rule R1) for a completely new variable V. Otherwise the details are routine. ∎

THEOREM 5.7 (UNIQUE SOLUTION OF EQUATIONS). *Let* $\tilde{X} = (X_1,..., X_m)$ *and* $\tilde{Y} = (Y_1,..., Y_n)$ *be distinct variables, and* $\tilde{F} = (F_1,..., F_m)$ *expressions with free variables in* $(\tilde{X}, \tilde{Y})$ *in which each* X_i *is guarded. Then there exist expressions* $\tilde{E} = (E_1,..., E_m)$ *with free variables in* $\tilde{Y}$ *such that*

$$\vdash E_i = F_i\{\tilde{E}/\tilde{X}\} \qquad (i \leqslant m).$$

Moreover, if the same property may be proved when $\tilde{E}$ *are replaced by expressions* $\tilde{E}' = (E'_1,..., E'_m)$ *with free variables in* $\tilde{Y}$, *then*

$$\vdash E'_i = E_i \qquad (i \leqslant m).$$

Proof. By induction on m. For $m = 1$ we choose $E_1 \equiv \mu X_1 F_1$, and the result follows immediately using the rule R4.

Assume the result for m, and now let $\tilde{F} = (F_1,..., F_m)$ and F_{m+1} be expressions with free variables in $(\tilde{X}, X_{m+1}, \tilde{Y})$ in which each X_i is guarded $(i \leqslant m + 1)$. We first find expressions $\tilde{E} = (E_1,..., E_m)$ and E_{m+1} such that

$$\vdash E_i = F_i\{\tilde{E}/\tilde{X}, E_{m+1}/X_{m+1}\} \qquad (i \leqslant m + 1). \tag{1}$$

For this purpose, first set

$$G_{m+1} \equiv \mu X_{m+1} F_{m+1} \tag{2}$$

$$G_i \equiv F_i\{G_{m+1}/X_{m+1}\} \qquad (i \leqslant m). \tag{3}$$

Since each G_i has free variables in $(\tilde{X}, \tilde{Y})$, with $\tilde{X}$ guarded, by induction there are expressions $\tilde{E} = (E_1, ..., E_m)$ with free variables in $\tilde{Y}$ such that

$$\vdash E_i = G_i\{\tilde{E}/\tilde{X}\} \qquad (i \leqslant m). \tag{4}$$

If we now choose

$$E_{m+1} \equiv G_{m+1}\{\tilde{E}/\tilde{X}\} \tag{5}$$

then we may first rewrite (4) using (3), then appeal to Lemma 5.6 and use (5) to obtain the required equations (1) for $i \leqslant m$; the details may be left to the reader. To obtain (1) for $i = m + 1$, we deduce from (2) and (5) (since X_{m+1} is not free in $\tilde{E}$)

$$E_{m+1} \equiv \mu X_{m+1}(F_{m+1}\{\tilde{E}/\tilde{X}\})$$

and hence by R2 and Lemma 5.6 (since X_{m+1} is not free in $\tilde{E}$)

$$\vdash E_{m+1} = F_{m+1}\{\tilde{E}/\tilde{X}, E_{m+1}/X_{m+1}\}$$

as required.

For the second part, suppose that (1) is satisfied also by expressions $\tilde{E}' = (E_1', ..., E_m')$ and E_{m+1}' with free variables in $\tilde{Y}$. Then first we have by Lemma 5.6

$$\vdash E_{m+1}' = F_{m+1}\{\tilde{E}'/\tilde{X}\}\{E_{m+1}'/X_{m+1}\}.$$

But X_{m+1} is guarded in $F_{m+1}\{\tilde{E}'/\tilde{X}\}$, whence by rule R4

$$\vdash E_{m+1}' = \mu X_{m+1}(F_{m+1}\{\tilde{E}'/\tilde{X}\})$$

and since X_{m+1} is not free in $\tilde{E}'$ we deduce

$$\vdash E_{m+1}' = G_{m+1}\{\tilde{E}'/\tilde{X}\}. \tag{6}$$

Second, we use this equation with the assumed property of $\tilde{E}'$ to obtain

$$\vdash E_i' = F_i\{\tilde{E}'/\tilde{X}, G_{m+1}\{\tilde{E}'/\tilde{X}\}/X_{m+1}\} \qquad (i \leqslant m)$$

and hence, by Lemma 5.6 and (3),

$$\vdash E_i' = G_i\{\tilde{E}'/\tilde{X}\} \qquad (i \leqslant m).$$

But from (4) using induction we obtain

$$\vdash E_i' = E_i \qquad (i \leqslant m).$$

Finally, summarising these equations by $\vdash \tilde{E}' = \tilde{E}$, we deduce from (5) and (6) that

$$\vdash E_{m+1}' = E_{m+1}$$

and the proof is complete. ∎

Before continuing towards the completeness theorem, we divert to prove the converse of Proposition 5.1. We adopt the convention that in an indexed sum $\sum_{j=1}^{n} E_j$, if $n = 0$ then $\mathbf{0}$ is intended.

COROLLARY 5.8. *If B is regular then it is expressible.*

Proof. Because B has a finite chart, there exist $B_1,\ldots, B_m$ with $B = B_1$, such that each B_i has a finite set $\{\langle a_{ij}, B_{f(i,j)}\rangle \mid 1 \leqslant j \leqslant m(i)\}$ of derivations, and a finite set $\{Y_{g(i,j)} \mid 1 \leqslant j \leqslant n(i)\}$ of extensions. Therefore, setting

$$F_i \equiv \sum_{j=1}^{m(i)} a_{ij} X_{f(i,j)} + \sum_{j=1}^{n(i)} Y_{g(i,j)} \qquad (m(i), n(i) \geqslant 0)$$

the following equations hold of behaviours:

$$B_i = F_i[\tilde{B}/\tilde{X}] \qquad (i \leqslant m).$$

But from the first part of Theorem 5.7 there are expressible behaviours $E_1,\ldots, E_m$ for which

$$E_i = F_i[\tilde{E}/\tilde{X}] \qquad (i \leqslant m).$$

(We are only concerned here with the truth, not the provability of these equations.) Now from the simple form of the expressions F_i, it is readily seen that $\{\langle B_i, E_i\rangle \mid 1 \leqslant i \leqslant m\}$ is a bisimulation of behaviours; whence $B_i = E_i$; hence B is expressed by E_1. ∎

Having shown in Theorem 5.7 that suitably guarded equations are provably satisfied by expressible behaviours, we proceed to show that, conversely, every expressible behaviour provably satisfies a set of equations.

THEOREM 5.9 (EQUATIONAL CHARACTERISATION). *For any expression E, with free variables in $\tilde{Y}$, there exist expressions $E_1,\ldots, E_p$ $(p \geqslant 1)$ with free variables in $\tilde{Y}$, satisfying p equations*

$$\vdash E_i = \sum_{j=1}^{m(i)} a_{ij} E_{f(i,j)} + \sum_{j=1}^{n(i)} Y_{g(i,j)} \qquad (1 \leqslant p) \text{ and moreover } \vdash E = E_1.$$

Proof. By induction on the structure of E. The only nontrivial case is $E \equiv \mu XF$. Now F has free variables in $(X, \tilde{Y})$, so by induction we have expressions $F_1,\ldots, F_p$ satisfying p equations which may be written

$$\vdash F_i = \sum_{j=1}^{m(i)} a_{ij} F_{f(i,j)} + \sum_{j=1}^{n(i)} Y_{g(i,j)}[+X] \qquad (i \leqslant p)$$

in each of which the summand X may or may not occur; also $\vdash F = F_1$. Now set

$$F_1' \equiv \sum_{j=1}^{m(1)} a_{1j} F_{f(1,j)} + \sum_{j=1}^{n(1)} Y_{g(1,j)}$$

so that either $\vdash F = F'_1$ or $\vdash F = F'_1 + X$. It follows, by rules R2 and R3, that

$$\vdash E = F'_1\{E/X\}. \tag{1}$$

Now set

$$E_i \equiv F_i\{E/X\} \qquad (i \leqslant p).$$

Then by the instantiation $\{E/X\}$ of the given equations we obtain, using (1) for any summand X,

$$\vdash E_i = \sum_{j=1}^{m(i)} a_{ij}E_{f(i,j)} + \sum_{j=1}^{n(i)} Y_{g(i,j)} \left[+ \sum_{j=1}^{m(1)} a_{1j}E_{f(1,j)} + \sum_{j=1}^{n(1)} Y_{g(1,j)} \right] \qquad (i \leqslant p)$$

which by rearrangement, are equations of the desired form. Moreover, $\vdash E = E_1$ follows from $\vdash F = F_1$, and the expressions E_i are easily seen to have free variables in $\tilde{Y}$. ∎

THEOREM 5.10 (COMPLETENESS). *If $E = E'$ then $\vdash E = E'$.*

Proof. Let E and E' have free variables in $\tilde{Y}$. By Theorem 5.9 there are provable equations $\vdash E = E_1$, $\vdash E' = E'_1$ and

$$\vdash E_i = \sum_{j=1}^{m(i)} a_{ij}E_{f(i,j)} + \sum_{j=1}^{n(i)} Y_{g(i,j)} \qquad (i \leqslant p)$$

$$\vdash E'_i = \sum_{j=1}^{m'(i)} a'_{ij}E'_{f'(i,j)} + \sum_{j=1}^{n'(i)} Y_{g'(i,j)} \qquad (i \leqslant p').$$

Now let $I = \{\langle i, i' \rangle \mid E_i = E'_{i'}\}$. Clearly $\langle 1, 1 \rangle \in I$. Moreover, since E_i and $E'_{i'}$ must have equal derivations and equal extensions when $\langle i, i' \rangle \in I$, the following hold:

(1) For each $\langle i, i' \rangle \in I$, there exists a total surjective relation $J_{ii'}$ between $\{1,..., m(i)\}$ and $\{1,..., m'(i')\}$, given by $J_{ii'} = \{\langle j, j' \rangle \mid a_{ij} = a'_{i'j'}$ and $\langle f(i, j), f'(i', j') \rangle \in I\}$;

(2) $\vdash \sum_{j=1}^{n(i)} Y_{g(i,j)} = \sum_{j=1}^{n'(i')} Y_{g'(i',j)}$.

We now consider the formal equations, one for each $\langle i, i' \rangle \in I$:

$$X_{ii'} = \sum_{\langle j,j' \rangle \in R_{ii'}} a_{ij}X_{f(i,j),f'(i',j')} + \sum_{j=1}^{n(i)} Y_{g(i,j)}$$

where the $X_{ii'}$ are not in $\tilde{Y}$.

First we assert that they are provably satisfied when each $X_{ii'}$ is instantiated to E_i. To see this, note that the typical equation becomes

$$E_i = \sum_{\langle j,j' \rangle \in R_{ii'}} a_{ij}E_{f(i,j)} + \sum_{j=1}^{n(i)} Y_{g(i,j)}$$

and is provable, since—as $J_{ii'}$ is total—its right-hand side differs at most by repeated summands from that of the already proved equation for E_i.

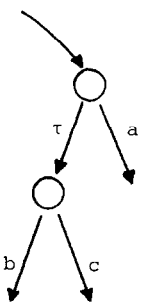
Second, the formal equations are provably satisfied when each $X_{ii'}$ is instantiated to $E'_{i'}$; this depends on the surjectivity of $J_{ii'}$.

Finally, we note that each $X_{ii'}$ is guarded in the right-hand sides of the formal equations. It immediately follows from Theorem 5.7 that $\vdash E_i = E'_{i'}$ for each $\langle i, i' \rangle \in I$, and hence $\vdash E = E'$. ∎

Remark. The main idea of this proof is essentially that of Salomaa's proof of completeness for an axiom system for regular expressions [9, Theorem 2]. However, our different interpretation of nondeterministic acceptors entails significant difference in detail, and it was by no means obvious to the author that Salomaa's idea could be adapted.

Let us briefly consider the introduction of empty transitions into charts, as mentioned at the end of Section 2. In the standard literature these are represented by arcs labelled ε (the empty word); in our own work [4, 7] the symbol τ was used since our interpretation is not in terms of words, and the transitions correspond to internal actions of a machine—arising perhaps from intercommunication of component machines. We cannot eliminate them as easily as can be done in standard automata theory.

Nevertheless, a corresponding notion of bisimulation can be defined, on the lines of the observation equivalence of Hennessy and Milner [4]. Intuitively, a sequence of τ-transitions is equivalent to a single one, but a single one cannot be totally ignored since it can have the effect of destroying certain capabilities of observation; in the diagram below, the ability to accept the symbol a can be lost



In the last-mentioned paper, a complete inference system for finite behaviours with τ-transitions was presented. It remains open whether the axioms given there can be combined with our rules to yield a complete system for regular behaviours with τ-transitions.

6. Star Behaviours

We now examine more closely the relation between behaviours and the standard theory of regular expressions and their interpretation.

To obtain a precise comparison, we first restrict attention to those behaviours in which at most the distinguished variable X is live; let us call them *X-behaviours*. Then each X-behaviour has an *X-chart*, one in which at most X appears as an extension, and which (if finite) may be interpreted as a finite-state acceptor in the standard way, by taking as accept states the nodes labelled X.

The regular operators ϕ, $+$, $\circ$ and $*$ over languages mean (in the standard interpretation), respectively, the empty language, union, concatenation and iteration (Kleene star). Over X-behaviours we give them the interpretation

$$\phi \mapsto \mathbf{0}$$

$$B_1 + B_2 \mapsto B_1 + B_2$$

$$B_1 \circ B_2 \mapsto B_1[B_2/X]$$

$$B^* \mapsto \mu Y(B[Y/X] + X).$$

To understand regular expressions over Act as X-behaviours, we need only add the interpretation

$$a \mapsto aX \qquad (a \in \text{Act}).$$

Now, to avoid confusion, let us rename the standard regular expressions over Act *star expressions*. Each star expression e may be regarded as an abbreviation for a behaviour expression E; for example, a abbreviates aX, and if e, f abbreviates E, F then $e \circ f$ abbreviates $E\{F/X\}$. We shall use the lowercase letters e, f, g to stand for star expressions; thereby they may also stand for the behaviour expressions which they abbreviate, or for their behaviours—which we will call *star behaviours*.

Salomaa [9] provides a complete inference system for star expressions under standard interpretation. When we dualise it, by writing $f \circ e$ for $e \circ f$ everywhere in Salomaa's rules (which gives an equipotent system), it has the following rules:

A_1	$e + (f + g) = (e + f) + g$	A_7	$e \circ \phi^* = e$
A_2	$(e \circ f) \circ g = e \circ (f \circ g)$	A_8	$e \circ \phi = \phi$
A_3	$e + f = f + e$	A_9	$e + \phi = e$
A_4	$(e + f) \circ g = e \circ g + f \circ g$	A_{10}	$e^* = \phi^* + e \circ e^*$
A_5	$e \circ (f + g) = e \circ f + e \circ g$	A_{11}	$e^* = (\phi^* + e)^*$
A_6	$e + e = e$		

R_2 *If* f does not possess e.w.p. then

$$\text{from } e = f \circ e + h \text{ infer } e = f^* \circ h.$$

(We have omitted R_1, the substitution rule.)

The empty word property (e.w.p.) is defined over star expressions by

(i) e^* has e.w.p.

(ii) If e or f has e.w.p. then $e + f$ has e.w.p.

(iii) If e and f have e.w.p. then $e \circ f$ has e.w.p.

If we regard e, f, g as X-behaviour expressions, then the following is easily established (we omit the proof).

PROPOSITION 6.1. (1) *e has e.w.p. iff X is not guarded in e.*

(2) *All except A_5 and A_8 of Salomaa's rules are derivable from the inference system of Section 5.*

The most important fact to notice in the correspondence between the two systems is that Salomaa's A_{10}, A_{11} and R_2 follow from our R2, R3, R4.

Moreover, A_5 and A_8 are not *valid* for behaviours. For from A_5 we derive $abX + acX = a(bX + cX)$, seen to be false in Section 2 (Example 3); from A_8 we derive $a\mathbf{0} = \mathbf{0}$, also false.

The question immediately arises whether, by removing A_5 and A_8, we obtain a complete inference system for star behaviours (it is certainly sound, by Proposition 6.1).

The answer is probably no for a trivial reason; the valid equation $\phi \circ e = \phi$ does not appear to be derivable (in deriving it, A_8 seems necessary). But we may add

A_8' $\quad \phi \circ e = \phi$

and the question may then be harder to answer. The difficulty is that the method of Theorem 5.10 or of Salomaa's original completeness proof cannot be applied directly, since—in contrast with the case for languages—an arbitrary system of guarded equations in X-behaviours cannot in general be solved in star expressions.

To see the difficulty informally, let us look at three examples. The first can be solved in star expressions, the second apparently cannot and the third certainly cannot, as we show formally later.

First, note that $\phi^* = X$ (equality of X-behaviours) easily follows from our definitions. This enables us to work directly with star expressions in attempting to obtain solutions.

EXAMPLE 1. Solve for f and g

$$f = a_1 f + a_2 g$$
$$g = b_1 f + b_2 g + X.$$

Translating into star expression notation, we obtain

$$f = a_1 \circ f + a_2 \circ g$$
$$g = b_1 \circ f + b_2 \circ g + \phi^*.$$

Now from the first equation

$$f = a_1^* \circ a_2 \circ g$$

whence by substitution in the second

$$\begin{aligned} g &= b_1 \circ a_1^* \circ a_2 \circ g + b_2 \circ g + \phi^* \\ &= (b_1 \circ a_1^* \circ a_2 + b_2) \circ g + \phi^* \end{aligned} \qquad (\dagger)$$

and finally

$$g = (b_1 \circ a_1^* \circ a_2 + b_2)^*.$$

EXAMPLE 2. Solve for f and g

$$\begin{aligned} f &= a_1 f + a_2 g + X \\ g &= b_1 f + b_2 g + X. \end{aligned}$$

Working as in Example 1 up to (†), we obtain instead

$$g = b_1 \circ a_1^* \circ (a_2 \circ g + \phi^*) + b_2 \circ g + \phi^*.$$

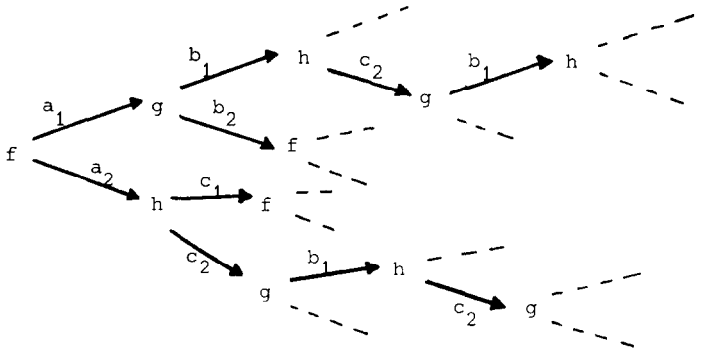
But now we need the (invalid!) distributive law A_5 to extract and collect the terms in g on the right-hand side. We cannot apparently continue.

EXAMPLE 3. Solve for f, g and h

$$\begin{aligned} f &= a_1 g + a_2 h \\ g &= b_1 h + b_2 f \\ h &= c_1 f + c_2 g. \end{aligned}$$

The same difficulty as in Example 2 occurs when we first eliminate f, say, and attempt to solve for g.

In the case of Example 3, Theorem 6.5 will show that the unique solution for f is not a star behaviour. The intuitive idea (slightly simplified) of the proof is that f, though infinite, possesses an infinite computation in which f never occurs as a proper derivative, and that the same fact is true of g and h, the only other distinct derivatives of f. Lemma 6.4 will show that this state of affairs cannot arise for a star behaviour. It is helpful to see the computations of f in tree form:



We first need a technical proposition, the analogue of a standard result.

PROPOSITION 6.2. *Every star expression e is behaviourally equivalent to one having no subexpression f^* for which f possesses e.w.p.*

Proof. We first show, by induction on e, that if e possesses e.w.p. then $e = \phi^* + e'$ for some e' without e.w.p. and with no greater depth of nesting of * than e. This follows simply from Salomaa's inference system without A_5 and A_8; for the crucial case $e \equiv f^*$ we appeal to A_{10} and A_{11}.

The result then follows by replacing each subexpression f^* of e, with f possessing e.w.p., by f'^* with f' not possessing e.w.p., working from the outermost. ∎

Now we look for a properly which is possessed by all star behaviours but not by all regular behaviours. The intuition behind the property which we exhibit is provided by Example 3 above. Roughly, we show that every infinite star behaviour must have some infinite derivative B_0, such that every infinite computation of B_0 reenters the "state" B_0 at some point. While not completely accurate, this remark should help the reader to understand the following definition.

DEFINITION. An X-behaviour B is *infinite* if it is possesses an infinite computation. B is a *loop* if, for some infinite B_0 and some C, $B = B_0 + C$, where every computation of B_0 has an initial segment

$$B_0 \xrightarrow{a_1} \cdots \xrightarrow{a_n} B_n \xrightarrow{a'} B' \qquad (n \geqslant 0)$$

for which

(i) $B_i \not\rightarrow X$, $0 \leqslant i \leqslant n$,

(ii) either $B' = \mathbf{0}$ or $B' = B_0 + C'$ for some C'.

LEMMA 6.3. *If B has a loop derivative, then so has $B + C$.*

Proof. Straightforward. ∎

LEMMA 6.4. *If e is an infinite star behaviour, then it has a loop derivative.*

Proof. By induction on the structure of e, which we may assume (by Proposition 6.2) has no subexpression f^* where f possesses e.w.p. Assume now that e is infinite.

(i) If $e \equiv \phi$ or $e \equiv a$ then it is a finite behaviour.

(ii) If $e \equiv f + g$ then f, say, is infinite; but since f has a loop derivative, so has e by Lemma 6.3.

(iii) If $e \equiv f \circ g \equiv f[g/X]$ then either f is infinite or f has a derivative $B + X$ and g is infinite.

In the first case f has a loop derivative $B_0 + C$, where B_0 satisfies the properties of the definition, so $B_0[g/X] + C[g/X]$ is an infinite derivative of $f \circ g$. But every computation of $B_0[g/X]$ has an initial segment

$$D_0 \xrightarrow{a_1} \cdots \xrightarrow{a_n} D_n \xrightarrow{a'} D'$$

with $D_i = B_i[g/X]$, $0 \leqslant i \leqslant n$, since its computations cannot divert from those of B_0 until a derivative with extension X is reached, and moreover

$$D_i \not\rhd X, \qquad 0 \leqslant i \leqslant n$$

$$D' = \mathbf{0} \qquad \text{or} \qquad D' = B_0[g/X] + C'[g/X].$$

Thus $B_0[g/X] + C[g/X]$ is a loop derivative of e.

In the second case e has a derivative $B[g/X] + g$, which has a loop derivative (since g does), so also e does.

(iv) If $e \equiv f^*$, where f does not possess e.w.p., then $e = f \circ e + \phi^*$.

If f is infinite the argument is as for the first case of (iii), appealing to induction for f. Otherwise f is finite, and $f \not\rhd X$ since f does not possess e.w.p. So each computation of f must have an initial segment

$$f = B_0 \xrightarrow{a_1} \cdots \xrightarrow{a_n} B_n \xrightarrow{a'} B' \qquad (n \geqslant 0)$$

where $B_i \not\rhd X$, $0 \leqslant i \leqslant n$, and *either* $B' = \mathbf{0}$ *or* $B' = X + C'$ (that is, after finite length either a computation ends or it meets extension X for the first time). Note that $B_0 \neq \mathbf{0}$, else e would be finite. Moreover the second alternative $B' = X + C'$ must occur for some computation, else f would have no derivative with extension X and again e would be finite.

We now assert that e is itself a loop. For it has the form $D_0 + \phi^*$, where each computation of $D_0 = f \circ e$ has initial segment

$$D_0 \xrightarrow{a_1} \cdots \xrightarrow{a_n} D_n \xrightarrow{a'} D'$$

with $D_i = B_i[e/X]$, $0 \leqslant i \leqslant n$, hence $D_i \not\rhd X$, and either $D' = \mathbf{0}$ or $D' = e + C'[e/X] = f \circ e + \phi^* + C'[e/X]$. Further, D_0 is infinite since the second alternative occurs for some computation, so e satisfies the loop conditions. ∎

THEOREM 6.5. *Not every X-behaviour is a star behaviour.*

Proof. Consider the X-behaviour f defined in Example 3. (In fact, not even X is live in f.) We assert that f itself is not a loop. For if $f = B_0 + C$, where B_0 satisfies the loop conditions, B_0 can only be $a_1 g$ or $a_2 h$ or f itself. In each case, B_0 has an infinite computation which is one of the following

$$B_0 \xrightarrow{a_1} g \xrightarrow{b_1} h \xrightarrow{c_2} g \xrightarrow{b_1} \cdots$$

$$B_0 \xrightarrow{a_2} h \xrightarrow{c_2} g \xrightarrow{b_1} h \xrightarrow{c_2} \cdots$$

and in each case no proper derivative (i.e., neither of g, h) has B_0 as a summand. This contradicts the loop conditions.

By symmetry, g and h are not loops. But f, g and h are the only distinct derivatives of f, hence f has no loop derivative. But f is infinite, so by Lemma 6.4 f is not a star behaviour. ∎

The condition of possessing a loop derivative, though necessary, is far from sufficient to ensure that a behaviour is a star behaviour, even if it is regular. The reader may like to check that Example 2, though apparently not a star behaviour, does satisfy the condition; take $f = B_0 + C$, where $B_0 = a_1 f$ and $C = a_2 g + X$.

We conclude the section by listing a few intriguing open questions about star behaviours, though perhaps their interest is limited compared with questions about regular behaviours.

(i) Is Salomaa's system without A_5 and A_8, but with A_8' and/or other axioms, complete for star behaviours?

(ii) What structural property of finite charts is necessary and sufficient for star behaviour?

(iii) Is there, for each n, a star behaviour using only one action which is not represented by any expression of star height less than n? The answer for the standard interpretation, regular sets of words, is no (see [10]); in the present case we conjecture that f_n is such a behaviour, where

$$f_1 = a^*$$

$$f_{n+1} = (f_n \circ a)^* \qquad [\text{e.g., } f_3 = ((a^* \circ a)^* \circ a)^*]$$

but have not found a proof. Note that $a^* \circ a \neq a \circ a^*$ in the behaviour interpretation.

ACKNOWLEDGMENTS

I would like to thank David Park for discussions about bisimulation, which greatly enhances the formulations and proofs in this paper.

REFERENCES

1. J. W. DE BAKKER AND J. I. ZUCKER, Denotational semantics of concurrency, *in* "Proceedings, 14th ACM Symposium on Theory of Computing," pp. 153–158, 1982.
2. B. COURCELLE, "Fundamental Properties of Infinite Trees," Report No. 8105, Uer de Mathématique et Informatique Bordeaux, 1981.
3. C. C. ELGOT, Monadic computation and iterative algebraic theories, *in* "Logic Colloquium" (Rose and Shepherdson, Eds.), pp. 175–230, North-Holland, Amsterdam, 1973.
4. M. HENNESSY AND R. MILNER, On observing nondeterminism and concurrency, *in* "Lecture Notes in Computer Science," Vol. 85, pp. 299–309.
5. C. A. R. HOARE, S. D. BROOKES, AND A. W. ROSCOE, "A Theory of Communicating Sequential Processes," Programming Research Group, Oxford University, 1981.
6. K. JENSEN, "A Method To Compare the Descriptive Power of Different Types of Petri Nets," DAIMI Report PB-108, Aarhus University, 1980.
7. R. MILNER, A calculus of communicating systems, *in* "Lecture Notes in Computer Science," Vol. 92, Springer-Verlag, New York/Berlin, 1980.
8. D. M. R. PARK, "Concurrency and Automata on Finite Sequences," Computer Science Department, University of Warwick, 1981.
9. A. SALOMAA, Two complete axiom systems for the algebra of regular events, *J. Assoc. Comput. Mach.* **13** (1) (1966), 158–169.
10. A. SALOMAA, "Jewels of Formal Language Theory," Computer Science Press, Potomac, Md., 1981.